



commodore
PET USERS CLUB
NEWSLETTER

Volume 2

Issue No. 1

Editorial

Welcome to Volume 2. Issue 1.!

The Pet Users Club is now in its second year of operation and is still growing. Membership is now around 2000 with more subscriptions coming in all the time.

This issue contains the first in a series of articles entitled "An Introduction to Machine Code", the most requested addition in the Pet Users Club survey. Also printed are two more reviews of Pet-related books.

Many thanks to the contributors to this issue; articles and programming routines are now trickling in steadily. Since we now have a lot of new members, it should be explained that the PUC depends on your contributions. If you can write anything about your application, programs or specialist hardware, please send it in.

Finally, this is the last issue for which I shall be Editor and Andrew Goltz will be taking over for the time being. The Pet Users Club will continue to be based at:-

360 Euston Road
London NW1 3BL.

Richard Pawson

Commodore News

NEW - COMMODORE TRAINING DIVISION

In keeping with our philosophy of continuous expansion and development of support activities, Commodore is about to commence a series of training courses on the use of Pet and its associated peripherals.

The courses are residential and of two or three days duration. A number of venues have been fixed, at different places across the country, in order to make the courses available to as many Pet users as possible.

All different types of user will be catered for, including ; business, scientific, educational and private use, though it is by no means necessary to own a Pet to benefit from them. The first two titles have been fixed as "Primary Basic" and "Disk Utilisation", with more to follow in 1980, such as "Advanced Basic" and "Interfacing".

There will be one Pet between each two students on the course, with small tutorial groups - a method of teaching already proven to be successful. Mike Gross-Niklaus (who has recently joined Commodore) will be supervising each course personally. Mike is well experienced at computer training, both with large systems and more recently with Pets in particular; he has also been contributing to this newsletter for some time now.

Overleaf is a list of dates and venues for the last quarter of this year. Please note that we can only accept official company orders. All other bookings must be accompanied with payment.

* * * * *

ENDORSEMENT SCHEME

This scheme, which was mentioned last issue, is now well established. You will probably have seen the "Commodore Approved" signs in your local dealer or magazine advertisements. Included with this newsletter you should have received a leaflet giving more details. Items already on the approved list include; interfaces, business suites, field maintenance contracts, books and firmware etc.

Commodore Training Courses — October to December 1979

Date	Course	Venue	Price With accom.	Without accom.
Oct 22-24	Primary Basic	Skyway, London	£225	£175
Oct 25-26	Disk Utilisation	Skyway, London	£125	£100
Oct 29-31	Primary Basic	N. Stafford, Stoke	£225	£175
Nov 1-2	Disk Utilisation	N. Stafford, Stoke	£125	£100
Nov 5-7	Primary Basic	Excelsior, Glasgow	£225	£175
Nov 8-9	Disk Utilisation	Excelsior, Glasgow	£125	£100
Nov 12-14	Primary Basic	Skyway, London	£225	£175
Nov 15-16	Disk Utilisation	Skyway, London	£125	£100
Nov 19-21	Primary Basic	N. Stafford, Stoke	£225	£175
Nov 22-23	Disk Utilisation	N. Stafford, Stoke	£125	£100
Dec 3-5	Primary Basic	Excelsior, Glasgow	£225	£175
Dec 6-7	Disk Utilisation	Excelsior, Glasgow	£125	£100
Dec 18-21	Primary Basic	Post House, Coventry	£225	£175

All courses start at 10.00 a.m. on Monday or Thursday.

Primary courses end at 3.30 p.m. on Wednesday.

Disk courses end at 12.30 p.m. on Friday. Prices are inclusive of VAT.

These are intensive small group courses involving full evening sessions and you are strongly advised to book *with* accommodation.

All accommodation is in top class Trust House Forte hotels where the actual courses take place. Single rooms with private bath and full board are standard. Bookings will be accepted on a first come basis.

Further courses on other topics will be run in the future. Full details of these and other PET developments will be published in the PET Users Club Newsletter, available for an annual subscription of £10.00 from the PET Users Centre, 360 Euston Road, London NW1 3BL.

BOOKING FORM

Please underline course required:

Basic Basic

Disk Programming

Date required: (1st choice) (2nd choice) (3rd choice)

Name(s)

.....
.....
.....

Address..... I enclose my cheque no:
..... for £ (or official
..... Company Order (no:))

Signature

Date

Please remit full payment or official company order — no cancellations can be considered within two weeks of the course starting date.

Please return to:
Commodore Systems PET Users Centre,
360 Euston Road, London NW1 3BL.

Software Notes

FUTURE TAPE RELEASES

Future additions to Commodore's cassette library will include programs on stock market trends, fast fourier transforms, logic circuit models, education in chemistry, analysis of cassette tapes and visual reports on any errors present, fantasy games and more besides. For full details of these new releases see the next edition of this newsletter and also our new master library catalogue (coming out shortly).

Our disk unit is now becoming widely accepted as a powerful addition to the Pet system. Because of the nature of its facilities and commands the disk requires careful use if it is to function correctly. Several users have experienced difficulties with the error light appearing on initialisation. This problem has been traced to incorrect centralisation of the media - the correct procedure is detailed below.

We are taking a number of steps to increase the documentation and education available on the disk - a new manual is being supplied with all subsequent shipments and printed elsewhere in this newsletter is a copy of our disk utility program.

DISK INITIALIZATION

The disk should be carefully centred in its envelope before insertion and most important of all, the door of the drive should be left open for 1 - 2 seconds as initialization proceeds. The effect of leaving the door open during the early stages of initialization is to allow the disk to settle into its optimum physical position before reads or writes are attempted.

In a few weeks we will be announcing the first release of software for the disk. Titles will include an Assembler, an Editor, LISP and the first of our Business Packages. These will include a Business Filing System and a Stock Recording with Invoicing suite and a Word Processor.

Ledger programs are complete and now being exhaustively tested.

As usual, may I ask you not to contact us about these products until an official announcement is made. Full descriptions of these products in the next issue and, of course, the results of the Star Trek competition.

The rest of this column is devoted to description of a fix for garbage collect problems.

NEWS from the... Petsoft Catalogue

Simulations and Games

NEW

BLITZ £5 A video game in which you command an anti-aircraft battery defending London. A direct hit on a falling bomb scores you 20 points, on a bomber 30 points. However, you lose 100 for every building destroyed.

NEW

BREAKTHROUGH £5 Classic video game with a moving gun shooting bricks out of a four layer wall; when you have made a hole through all four layers, you shoot one more bullet through the hole to win. Three speeds of play, each with a limit on the number of shots. N.B. Fast speed is very difficult but achievable. Program can be modified to vary speed and wall thickness.

NEW

BULLS & BEARS £9 Jeff Boetticher, Head of the American Byte Shop chain, reckons this is the best micro computer game available anywhere. Each player starts with \$50,000 to invest on the stock market. By manipulating shares in the companies you control and keeping a tight rein on the production side, it is possible to become a millionaire. Revised and improved U.K. version.

NEW

HEXAPAWN & FRAMEDART £5 HEXAPAWN is played with two sets of three pawns on a 3x3 board, the object being to reach the other side or to prevent the computer from moving. Be warned — PET learns from its mistakes and improves its play each game. FRAMEDART generates a constantly changing screen pattern to relax you after Hexapawn.

NEW

HUNT £10 A new concept in fantasy simulations which has achieved wide acclaim. "The context is that of a search for a defined object, typically Atlantis or the Holy Grail. The objective, the names and natures of the searchers, their antagonists and the properties of the space in which the hunt is conducted are defined — by you!" — Practical Computing.

NEW

MAXITI £5 An ingenious board game played on 8x8 board. You can only move horizontally whilst PET is allowed vertical moves only. Warning: PET's strategy is very cunning.

NEW

THREE OF A KIND £4 The computer selects nine words. You and PET take turns to pick one out, the object being to get three of the same letters into your word selections. A real brain boiler!

NEW

SUB SEARCH £4 A new submarine-hunting game with dynamic graphics. Save the convoys by depth charging enemy U-boat

NEW

THE MONEY GAME £5 Your chance to invest on the stock market and the turf. Fortunately, rich uncles die regularly leaving you shares. You may get away with not declaring your gains — or you might be arrested!

NEW

MOTOR RACING £6 Exciting race track simulation. Controls include accelerator, brake and steering. Good graphics.

NEW

PARTY PACK £5 Children love this package which includes GUESS THE NAME, where your clues include the first letter and sex of the name, and SNAP with variable speeds. CONSEQUENCES displays a random series of sentences describing who meets who, what they do, and the results. The sentences can be personalised to intriguing effect.

NEW

PET POETRY £5 We think the poetry generated from PETs data files — which you can easily amend — is as good as most of the rubbish that gets published nowadays.

NEW

SCRAMBLE The computer generates anagrams of well known words for you to unravel. Not nearly as easy as it sounds. Crossword puzzle addicts will adore this.

NEW

SOLITAIRE £4 Now you can play this famous game on the PET. If you make a mistake, cheat a little and have the computer back step a few moves.

NEW

SPACE SHUTTLE £5 Can you manoeuvre your NASA shuttle to dock with an orbiting space station before you run out of fuel? Highly rated by all computer space aficionados.

NEW

STEEPLECHASE £5 All the thrills of the National Hunt. Meet 'Honest Joe' the punter's friend, and lose your shirt on the horses. Very good graphics. Start with £1000 and take your chance.

NEW

TILES £5 A challenging graphic game which requires you to move one large square amongst a number of smaller squares from one side of the board to the other and back. It takes PET 138 moves — can you do better?

Educational Mathematical and Scientific

NEW

BIOLOGY MULTIPLE CHOICE £6 C.A.I. program tests and stimulates knowledge of: Bone Structure of Birds, Calories, Bone Structure of Humans, Plant Structure, Protein in Urine, Eggs and Food, Glucose in Blood, Calorific value of Fats in Blood, Temperatures and Enzymes, Parts of the Plant, Human Body Temperature, Fungus, etc.

NEW

CALCULATOR £5 Comprehensive simulation converts numeric keypad into a four function calculator with memory and automatic constant. Features include fixed or floating decimal point, with round off or down. A calculator display is shown on screen and all steps are listed. Alternatively, a printer may be used to turn the PET into a powerful printing calculator.

NEW

CHEMISTRY TUTOR £7 Comprehensive tutorial dealing with atomic numbers, element symbols etc. Choice of difficult or easy questions.

76 COMMON BASIC PROGRAMS £15 A collection of 76 useful programs on one cassette from Adam Osborne's best selling book. These include personal finance, maths, statistics and general interest topics. Excellent value for stand alone or incorporated use.

NEW

COURSE HANDLER £95 A must for School Timetablers. The program handles all the information relevant to creating a 4th/5th or 6th year Option Scheme and is particularly useful where an "Open Choice" of subjects is offered to pupils. The program maintains, via a simple dialogue with the timetabler, a file of pupils and their requests and allocations and a file containing details of the Option Scheme. Facilities are provided for viewing the scheme, the classes, the pupils and the class clash matrix. The COURSE HANDLER will fit the pupils against the scheme by 'Best Manual Package' method but these assignments may be overridden by the timetabler if they are unsuitable. Further, pupils may be selected for editing by class or name for final adjustments to their choices. Described by users as "A great asset" and "A very useful tool". Suitable for a PET with 32K bytes and two cassette recorders. Full 65 page manual supplied.

NEW

CRYPTO PACK £8 This is the complete kit for all those interested in cryptography, codes, ciphers and cryptanalysis. Developed by Dr Michael Richter, the package includes Cryptosub, General Cipher, Cryptanalyser and New Cipher programs.

NEW

FREQUENCY RESPONSE £5 You enter the Constant, Finite Delay in L, Sec, number of zeros of transmission function. Frequency response is radians/second. Range option: octaves, halfoctaves or equal octaves. The program calculates cycles/second, radians/second, amplitude ratio in decibels and Phase in degrees, from the information entered. Print out function included.

NEW

MATHADAPT £5 Written by 'Kilobaud' PET columnist, Greg Yob, to develop basic maths skills, the program sets problems for Addition, Subtraction, Multiplication and Division, and marks a report card. However, this program gives you control on both problem limits and their difficulty.

NEW

PHOTOGRAPHY TUTOR £12 A comprehensive course developed by a professional photographer making full use of PET's dynamic graphics capabilities to demonstrate and explain the mysteries of exposure, focus, aperture, shutter speeds, interchangeable lenses, depth of field, etc. The theory and practice of photography are explored interactively, and progress tested. Multiprogram pack. Available on Disk £15.

NEW

PHYSICS MULTIPLE CHOICE £7 Clearly written program which tests and scores knowledge of electricity, Amps, Ohms, Volts, Galvanometers, Lenses, refraction of light, colours, magnetism, velocity, Newtons, Pulleys, measurement of humidity, convection, pendula, cathode rays, etc.

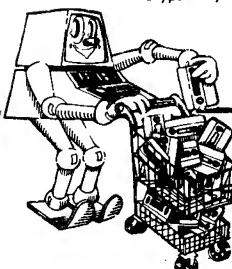
NEW

TRANSACTIONAL ANALYSIS £7 An introduction to T.A.—a system for understanding human behaviour. Chapters include: You As A Person, Stroking, Transactions, Are You Listening?, The Balancing Game. This interactive learning cassette will help you gain a better understanding of why you get along with some people and not with others, and may give you a better understanding of yourself! Documentation included.

NEW

FRENCH/ENGLISH TUTOR £6 The computer tests your language skills from English to French and French to English. Answers included, of course.

Continued



ACT Petsoft LTD.

Radclyffe House, 66-68 Hagley Road,
Edgbaston, Birmingham B16 8PF. Tel: 021-455 8585 Telex 339396

NEWS from the... Petsoft Catalogue

Business Programs

NEW

HOUSEFINDER £25 Professional package developed specially for Estate Agents. Details of properties on the register are encoded under Aea (up to 34 are handled), type of house, situation, No. of reception rooms, bedrooms, garden size etc. Modes include Enter or Display Details, Find Property, Save Files, Load Files, etc.

NEW

JOB EVALUATION £25 Conducts the evaluation necessary to establish pay structures and grades. Program computes correct weightings for factors — education, training necessary, responsibility over other men and equipment, working conditions etc. — which comprise job value. A Job Evaluation Formula is created for use as a guide to the relative value of a job *based on the thinking of the company or department concerned*. Answers to paired comparison tests with other jobs are analysed. The names of the factors for which weightings are given, are then requested and the comparison procedure activated. Multiple Regression Analysis is invoked to find the set of weightings best fitted, and display them in the form of a Job Evaluation Equation. Full documentation included.

NEW

ROUTE PLANNERS: France, Spain, Italy, Western Europe Trunk Routes — £5 each. Enter your present location and destination and PET searches its database for the most efficient route. The names of towns, distances etc. are displayed.

NEW

PAYROLL — 400 (Disk) £50 A totally new and complete disk based payroll system designed and written to meet the needs of small businesses. Up to 400 employees per disk are catered for. A 32K PET 2001-32 equipped with Commodore dual floppy, an Annadex DP800 or device 4 printer is required. Ease of use, security and versatility are key points in this system. Facilities provided include Holiday Pay, Sick Pay, Bonus payments and two rates of overtime, as well as allowing a "standard week" to be specified for each employee. Weekly and monthly summaries are provided and amendments necessary because of a Budget (e.g. increasing employee's tax codes) are made very easy. Each week a wage slip is printed for each employee followed by an analysis of the coins/notes required for these employees paid in cash (payments by cheque and credit transfer are also allowed for). Tax and N.I. are computed automatically from a knowledge of the tax code and N.I. rate applicable to that employee. All tax codes and N.I. tables A, B and C are catered for. As well as being simple and time saving to use, the system is designed to be "operator proof" and the security prevents unauthorised amendment of details on file. Versions of this system for other disk drives and printers will be available shortly. Update service available.

NEW

STOCK CONTROL ON DISK £25 Facilities allow full or operational stock print out, total costing of items in stock and re-order level warning. Data is stored under Reference, Description, Supplier, VAT Rates, On order Quantity, Quantity in Stock, Unit of Quantity Designated, Minimum level, Stock allocated, Sale Price and Purchase Price. Approx 400 items per diskette.

Programming Aids and Tutorials

NEW

AUTO LINE NUMBERER £5 Write your programs without bothering about line numbers. Automatically supplies the next line number when you hit Return. Increment and starting point specifiable by user. May also be used for deleting blocks of lines.

NEW

BUTTERFIELD'S ENCYCLOPAEDIA £12 A treasure trove of utilities and games from the PET's leading exponent, Jim Butterfield. Includes Copycat, to prevent tape copying, Tape Test, Morse Practice, Triangle, Calculator, Metric Conversion, Data Finder, Trendline, Mileage, Factors, Keyboard Record, Battleships, and many more. Outstanding value.

NEW

6602 FORTH £30 FORTH is a unique threaded language that is ideally suited for systems and applications programming on a PET. The user may have the interactive FORTH Compiler/Interpreter system running stand-alone in 8K to 12K bytes of RAM. The system also offers a built-in incremental assembler and text editor. Since the FORTH language is vocabulary based, the user may tailor the system to resemble the needs and structure of any specific application. Programming in FORTH consists of defining new words, which draw upon the existing vocabulary, and which in turn may be used to define even more complex application. Reverse Polish Notation and LIFO stacks are used in the FORTH system to process arithmetic expressions. Programs written in FORTH are compact and very fast. System features and facilities include Standard Vocabulary with 200 words, Incremental Assembler, Structured Programming Constructs, Text Editor, Block I/O Buffers, Cassette Based System, User Defined Stacks, Variable Length Stacks, User Defined Dictionary, Logical Dictionary Limit, Error Detection and Buffered Input.

NEW

SYSTEM EXTENSION £12 Allows you to store up to 10 separate Basic programs in RAM at once for instant individual retrieval. Includes Block Delete program linker and Fast Line Renumberer. Crash prevention feature with special error messages. Suitable for old ROM PETs with expansion memory only.

NEW

FORMATTED LISTER £15 Intelligible listings at last! Produces neat listings of Basic programs, which differ from the normal one in that the line numbers are right aligned, the listing is divided into numbered titled pages, and Basic lines too long to fit in the print width have their continuations indented clear of the line numbers. PET cursor control characters are printed as bracketed abbreviations e.g. (CLR) for "Clear Screen". For printers without lower case, shifted PET symbols are printed as bracketed upper case symbols. A blank line is printed as each line started with a REM statement. This allows separation of program line blocks.

NEW

MULTI KEY GET £4 How to make the PET recognize more than one key when several are pressed at once — essential for games. A demo is included in addition to the routine itself.

NEW

PILOT £10 The latest version of an additional language for the PET, especially suited to writing conversion programs and for Computer Aided Instruction. 15 easy-to-learn one-syllable commands handle the display of string, input commands, checking data against inputs, jumps to other program segments or subroutines, returns from subroutines, special counters etc. The PILOT interpreter also accepts fundamental Basic commands including LOAD, RUN, NEW, STOP, SAVE Etc.

NEW

SCREEN STORE £4 Useful routine for storing the exact contents of the screen RAM. Can also be used with dynamic graphics.

NEW

SELF MONITOR £7 An essential debugging aid which allows you to see your own program execute line by line. Variables below line 999 are also checked. The routine can be automatically deleted from your program.

NEW

SOFTWARE MATHS ENHANCER £9 Software method of enhancing the maths capabilities of the PET, includes: RPN Calculator, Hex Arithmetic, Fourier Explicit, Non Unif, 78 Digit Multi, 255 Digit Multi, Small Primes and Large Primes.

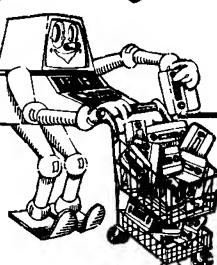
HOW TO ORDER

From your local PET Dealer,
or direct from us:

We accept Access, Mastercharge, Barclaycard, Visa, Eurocard orders by telephone on 021-455 8585. Official orders are accepted from schools, colleges, government departments only. All other orders should be accompanied by a cheque.

PLEASE ADD V.A.T. + 50p Post and Packing to your Order

Non-UK Sales We will, and do ship to any part of the world by airmail post. To cover the cost please add the equivalent of 25 pence (50 US cents) per title to destinations within Europe, and 50 pence (1 US\$) per title elsewhere. All orders should be prepaid. We accept Mastercharge, Eurocard, Bank Americard/Visa credit cards, cheques drawn on UK banks, and International Money Orders. Giro or Direct credit transfer may be made to our account.



RCT Petsoft LTD.

Radclyffe House, 66-68 Hagley Road,
Edgbaston, Birmingham B16 8PF Tel: 021-455 8585 Telex 339396

STRING FIX

There is a PET peculiarity (which is shared by all Microsoft Basic Machines), which can cause many problems - some of which might be thought to be due to the disk but are in fact due to the way strings are organised.

This is the problem of garbage collection which can cause a machine to "hang-up" or become unresponsive to the operator for long periods of time e.g. more than 130 seconds for the program below without the String Fix (which is in line 55).

From the tabulated results, it would seem strings are most quickly handled in this case with the top of memory set at N=23 (5.75K).

So, if long delays are encountered in programs handling strings, the remedy is to lower the top of BASIC to create a shorter stack.

Previously assigned strings may still be referenced, but all new assignments will now only be garbage collected to the new top of BASIC.

This can produce spectacular improvements in the performance of programs as the table above the listing suggests.

SECONDS	TOP OF BASIC (HI BYTE)
12.05	15
7.61	16
6.08	17
5.31	18
4.95	19
4.73	20
4.46	21
4.40	22
4.13	23
4.25	24
4.63	25
7.28	26

STRING FIX DEMO

```
5 POKE53, 128:REM SETS TOP OF MEMORY TO 32K
10 X=720
20 OPEN3,4:OPEN4,4,1:OPENS,4,2:PRINT #5,"99.99"               99"
30 DIMA (X)
40 FORI=ITOX
50 A$(I)=STR$(I) + "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
60 NEXT
70 PRINT#3, "SECONDS      TOP OF BASIC (HI BYTE)"
80 FORN=15TO26
90 POKE3,N
100 T=TI
110 FORJ=1TO800
120 Z$= "ZZZZZZZZZZZZZZZZZZ" + ""
130 NEXT
140 PRINT (100* (TI-T)/60)/100,N
150 PRINT#4, (TI-T)/60,N
160 NEXT
170 CMD4:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT" STRING FIX DEMO"
:PRINT:
PRINT:PRINT:LIST
READY.
```

PAYROLL

We are producing a Payroll 2 in response to new regulations and recommendations that have recently come from the Inland Revenue.

This program will be supported by a different author to the last one and we will be announcing details soon.

All orders for Payroll received after September 1st, will be held until they can be supplied with Payroll 2.

Hardware

THE BASIC PROGRAMMERS TOOLKIT - A REVIEW

The Basic Programmers toolkit is a piece of firmware, developed specifically for the PET, in the States, and imported into this country by PETSOFT. Essentially the toolkit adds 10 commands to the PET operating system by means of a PROM containing a number of machine-code subroutines residing in the expansion ROM area of the memory map.

For the new 16 & 32K Pets, this PROM will plug straight into one of the expansion ROM sockets on the main logic board. For 8K Pets, the toolkit is mounted on a small driver board which plugs into Pet's expansion memory port and 2nd cassette port. After connecting and switching on, the toolkit is activated with one SYS command.

The features added by the toolkit are mainly used in the development and de-bugging of programs - hence the name. Although one or two of the routines are already available on cassette from other sources, the advantage of the toolkit is that they are constantly available and do not use up your working RAM.

The toolkit comes with extensive documentation, including a list of 'GOTCHAS' (situations in which mis-application of the command results in errors).

The following is a list of the commands, with a brief explanation and some personal review comments. One nice feature is that all these commands can be abbreviated to two letters, as per Pets Basic commands.

AUTO

This command automatically prints the line numbers for you when keying in a program. The start line (defaults to 100 if not specified) and increment (defaults to 10) can be specified. After a line of BASIC is keyed in the next line number appears automatically.

Will save time, but not a lot.

DELETE

A real time saver - DELETE 300 - 500 will remove lines 300 - 500 inclusive. The command string can be shortened to DELETE 200 - etc. as in LIST. DELETE is especially useful at the early stages in a program when different routines are being tried.

FIND

Absolutely invaluable if you wish to change or alter a program's construction. FIND will search out and print all lines containing a specified string of characters and/or words. For example, FIND GOSUB 1000 will print all the lines which contain GOSUB 1000, and FIND A1 will list all lines in which the variable A1 is used.

RENUMBER

This will renumber your program, starting from a specified line number (defaults to 100) in equal specified increments (defaults to 10). All GOTO's and GOSUB's are changed accordingly. Although a renumber facility is useful for 'opening out' a crowded program, it is not possible to renumber a small part of a program. This means that if you start your subroutines at multiples of 1000 (1000, 2000, etc.) for an easily identifiable program structure, RENumber will lose this - very annoying.

APPEND

Adds a previously saved program onto the end of a program in memory. Although APPEND will not interleave line numbers as will the Butterworth method, this is not a great disadvantage since APPEND is mostly used for building up programs from a library of subroutines. It does have the advantage over the Butterworth method in that it appends normal saved programs and not pseudo data files.

DUMP

Prints out the names and values of all (non-dimensioned) variables used in a program, when it has stopped. Variables are displayed in their order of creation and can be changed with screen editing. A useful feature for finding out why a program is not working as expected.

HELP

When a program halts due to an error, HELP displays the appropriate line, with the offending character highlighted in reverse field. A useful teaching aid for beginners in Syntax Errors and a time saver for advanced programmers, especially when using multiple statements per line.

TRACE, STEP, OFF

These commands are used while a program is running and display the line numbers as they are executed in a reverse field window on the screen, together with the previous five line no.'s. The speed of operation can be controlled with the SHIFT and STOP keys and STEP allows Basic to be executed one line at a time. OFF turns TRACE and STEP off. These commands are somewhat disappointing in that they don't display the contents of the line as well as the line no.'s (unlike one tape-based version from the States) but do at least display several line numbers at once.

Programmers Toolkit is a Commodore Endorsed product and will be available from Petsoft in October. The chip for 16/32K Pets costs £55 + VAT and the version for 8K Pets - £75 + VAT.

DIGITAL TO ANALOGUE CONVERSION

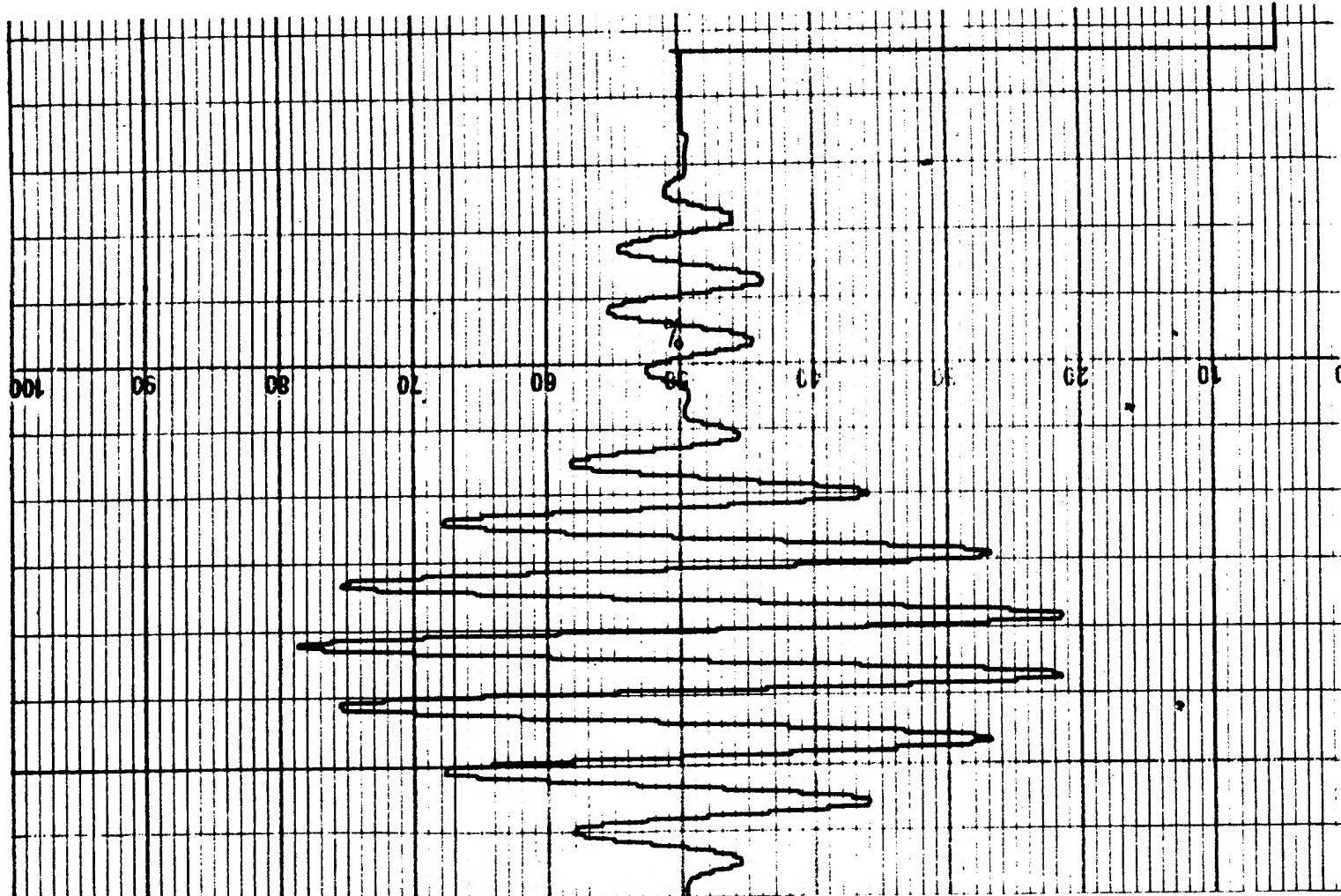
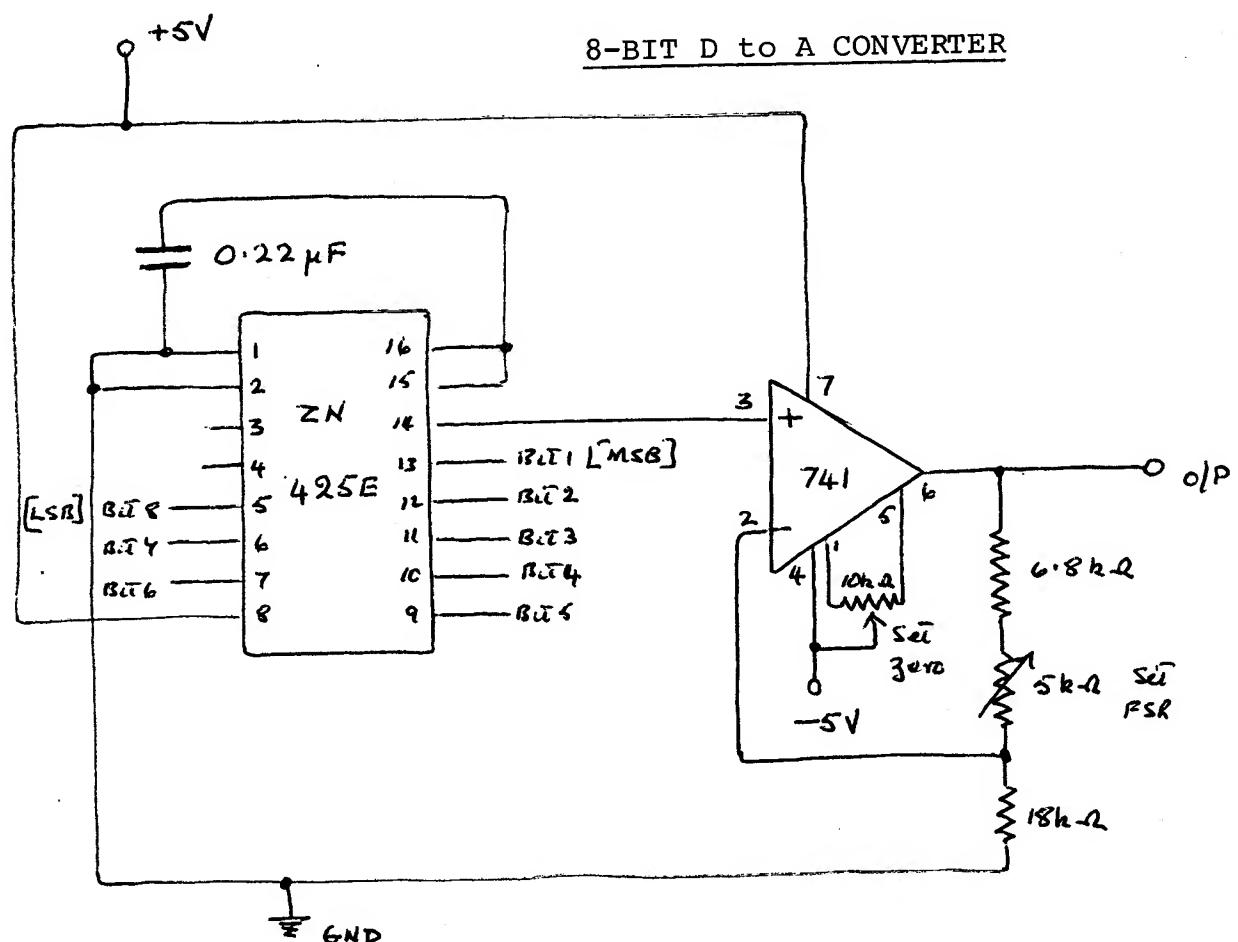
D. Muir of Napier College, Merchiston, Edinburgh, who sent in a design for the 'Blinkin' Lights' machine, published last issue, has sent us another circuit - this time for an analogue output from the Pet.

Designed to attach to the User Port or to the 'Blinkin' Lights' machine directly, the circuit is especially suitable for driving a "servoscribe" pen-recorder as mentioned by Dr. Smyth in issue No. 4. A sample print-out from this machine, along with the program to produce that waveform, is shown.

It should be noted that bit 1 on the D to A convertor must be connected to the Most Significant Bit on the User Port.

```
2000 POKE59471,128
2500 FORJ=1TO5000:NEXTJ
3000 K1=2*π/8:K2=K1/8
4000 A=A/5000
5000 DIMA(256),A%(256)
7000 FORI=0TO255:X=(I-128.001)/1.28
8000 A(I)=10*CO S (K1*X)*SIN(K2*X)/X
9000 A%(I)=128*A(I)
10000 PRINTI,X,A%(I)
10500 A%(I)=A%(I)+127
11000 POKE59471,A%(I)
11500 FORJ=1TO500:NEXTJ
20000 NEXTI
25000 FORJ=1TO5000:NEXTJ
30000 POKE59471,0
```

8-BIT D to A CONVERTER



INTERFACE PRODUCTS

IEEE-488/RS232C SERIAL INTERFACES

* Full IEEE address decoding for Pet disk compatibility*	
* RS232C or 20mA Loop output	* Crystal controlled BAUD rates*
* Custom chips allow any character code sets*	
* RS232C DTR (Printer Busy) input*	
Serial Interface B, input and output	186.00
Serial Interface C, output only	120.00



IEEE-488/CENTRONICS TYPE PARALLEL INTERFACES

ADDRESSABLE:-

* Full address decoding for Pet disk compatibility *	
* Custom chips allow any character code sets *	
* Available for CENTRONICS, ANADEX, PR40 and BD80 printers*	

Parallel Interface AD (addressable)	106.00
---	--------



NON-ADDRESSABLE:-

Low cost compatible with most parallel printers	45.00
---	-------

MICROPROCESSOR BASED GENERAL PURPOSE INTERFACE (G.P.I.)

GPI AP Version of GPI programmed for Pet interfacing applications.

* RS232C and 20mA Loop bidirectional	* RAM for input/output buffering *
* Software controllable BAUD rates	* 2 serial I/O ports capable
of asynchronous or synchronous operation	* Full RS232C handshake
capability	* Special output formatting facilities
for modem operation *	P.O.A.



TV/VIDEO MONITOR INTERFACE



Video a.o UHF output/plugs into aerial socket of domestic TV set	35.00
---	-------

COMPUTASTORE SOFTWARE



PETE Intelligent Terminal Software Package	100.00
--	--------

PET MEMORY BOARDS



Internally mounting - available with Prom sockets: 24K.....	328.00
32K.....	432.00

** NEW ** REAL-TIME AUDIO SPECTRUM ANALYSER

* Internally mounting	* 32 Channels	* 1K ROM routines on board for analysis and graphical display	* USR functions linkage to Pet operating system	450.00
-----------------------	---------------	--	--	--------

Prices EX VAT. Post & Package (Securicor express delivery) 5.00
All goods supplied under 90 days warranty

Beginning Machine Code — An Introduction

The most common request in our PUC survey, two issues ago, was for a series of articles explaining Machine Code from the absolute basics. In this issue, we shall be dealing with definitions by way of answering some commonly asked questions. For those of you who want to proceed faster than this, we recommend books such as Rodney Zaks "Programming the 6502".

* * * * *

WHAT IS MACHINE CODE? (abbrev. M/C code)

Machine code is the language understood by the Microprocessor at the heart of any micro-computer system. Each type of microprocessor requires its own special machine code, although most have several features and principles in common. This series will teach 6500 Machine Code since the Pet is based on a 6502 Microprocessor.

Any software to be run on a microprocessor must first be translated into machine code. Most personal computer software is written in a 'high-level' language such as BASIC (the higher the level of a language, the more it reads like English - machine code is termed a 'low-level' language since it consists of numbers or binary digits only).

There are two methods of translating a high-level language into M/C code. A compiler translates the whole program in one go, so that the program can then be run straight on the microprocessor, very efficiently. This, however, requires a lot of RAM space and so compilers are rare on micro-computers. The alternative is to use an Interpreter (as on the Pet) where the language is translated line by line and M/C subroutines in ROM are called as and when required by the Interpreter. As might be expected, the latter means that the program runs many times slower (particularly in loops), but needs less space, and is particularly suitable for interactive systems such as the one on your Pet.

WHAT ARE THE ADVANTAGES OF WRITING DIRECTLY IN MACHINE CODE?

If you can write a program (or part of a program) directly in M/C code then it will execute up to a hundred times faster. The disadvantage is that M/C code is rather more difficult to write than BASIC. It is thus common practice to write most of a large program in BASIC and 'drop down' into M/C code where a routine can easily be written thus.

WHAT TYPE OF ROUTINES ARE SUITABLE FOR WRITING IN MACHINE CODE?

Any routine which is largely made up from PEEK and POKE statements is an ideal candidate for M/C code writing. These include software interfaces to drive peripherals and routines for shifting blocks of memory.

Also suitable are routines which involve a simple operation being repeated a large number of times, such as string processing.

In contrast, routines which involve complex mathematical calculations are difficult to write in M/C code, unless you have access to a well written library of M/C code arithmetic subroutines.

HOW DO I RUN A MACHINE CODE ROUTINE ON THE PET?

There are two methods of accessing a M/C code routine from BASIC. SYS (location) will transfer the operation of PET from BASIC to the M/C code starting at the specified location. USR (parameter) will transfer to M/C code at a pre-determined location. In addition, it is possible to pass a parameter between BASIC and M/C code - either a BASIC variable or a constant. The operation of these two commands is detailed in the Pet manual.

WHAT DOES MACHINE CODE LOOK LIKE?

True M/C code, as stored in the memory, consists of binary digits. For compactness in print, ease of entry and to help avoid errors, the M/C code is often written in base 16 known as Hexadecimal or Hex. A piece of M/C code might be written thus:-

```
AD $3 IF
18
69 $4
8D $3 IF
$0
```

This kind of program is very difficult to read so the various instructions are given mnemonic codes. For example, the instruction 8D puts the contents of the accumulator into a particular byte of memory. This is given the mnemonic "STA" (store accumulator). The above program in this form might be written:-

MNEMONICS

```
LDA VALUE
CLC
ADC #$04
STA VALUE
BRK
```

EXPLANATION

```
Load "value" into accumulator
Clear contents of carry flag
Add with carry 4 to accumulator
Store accumulator in "value"
Break (halt)
```

This is called an Assembly Language program. Generally speaking, a routine is developed in Assembly Language. It is then converted to pure M/C code by means of an "Assembler". This is not as complicated as interpreting a high-level language because each mnemonic can be converted by a look-up table. This sort of conversion is known as "one to one".

A Disassembler converts M/C code into mnemonic form to help de-bug or alter an existing routine.

WHAT DO THE STATEMENTS IN A MACHINE CODE ROUTINE ACTUALLY DO?

The microprocessor contains certain 8-bit registers such as the Accumulator, X register and status register. It understands 56 types of instruction (e.g. ADC) although most of these have several variations or addressing modes. Most of these instructions perform a simple operation on one or more of these registers. For example, CLC sets one of the bits in the status register to zero. On a microprocessor the most advanced type of instruction is the addition or subtraction of two bytes. All other functions must be built up from this level.

Programming

We received a letter from Jim Butterfield, in Toronto, detailing the finalised version of the merge routine which we printed in a previous issue. The text follows:-

Merging PET programs: a final report

Jim Butterfield, Toronto

To wrap up the various activities surrounding merging or UNLIST, and bring them up to date with information on new ROM:

I. To change a program into a data file on cassette tape:

Mount blank tape on cassette 1. Type:

OPEN 1,1,1 : CMD 1 : LIST

Cassette tape will write. When writing is complete, the flashing cursor will return, but PET will not print READY - the word READY is in fact written on tape. Now close the CMD and tape file with:

PRINT #1 : CLOSE 1

This "merge" tape may now be saved for any future occasion.

Variations:

- the file may be named, e.g. OPEN 1,1,1, "TEST MERGE": ... etc.
It's good practice to name files if you plan to keep them.
- if desired you may copy only part of the program to tape,
e.g. ... CMD 1 : LIST 500-700 ... This is a handy way to extract subroutines from a larger program.

II. To merge a data file (in the above format) into program space:

The procedure is slightly different on original ROM as compared to the new ROM, which I'll call upgrade ROM.

The program with which you wish to merge must first be loaded into memory. The following procedure may be repeated many times, so that you may merge several program blocks together.

Mount "merge" tape on cassette 1. Type:

Original ROM: POKE 3,1 : OPEN 1
Upgrade ROM: POKE 14,1: OPEN 1

Tape will now be read. Eventually, the computer will report FOUND and the cursor will return.

Now: clear the screen and press exactly three cursor downs. Type:

Original ROM: POKE 611,1 : POKE 525,1 : POKE 527,13 : ?"h"
Upgrade ROM: POKE 175,1 : POKE 158,1 : POKE 623,13 : ?"h"

("h" is the cursor home key - it will print as a reverse S).

As soon as you press RETURN at the end of this line, the word READY will appear above the line, and tape will move. When the merge is complete, the computer will print either ?OUT OF DATA ERROR or ?SYNTAX ERROR below the line. This is normal and does not signify a real error. The job is now complete.

Note the four new items:

- - a new POKE statement before OPEN 1;
- - three cursor downs before the final POKE;
- - only one final POKE line to be typed;
- - no need to close the file at end of merge.

The new system is simpler, and also corrects a minor problem on the original POKE 611 merge. Few people spotted it, but the original procedure caused line 1 to disappear.

* * * * *

The following routines were developed by Paul Higginbottom for use in the Commodore Pet Centre.

The first, which is for 8K Pets but can be adapted for new Rams, allows repeated cursor movements. The cursor in this routine is shown as an asterisk and will move in the desired direction as long as the key is pressed down:

```

10 PRINT "ch ↓ ↓ ↓ ⇒ ⇒ ⇒ *← ;
20 A = (PEEK (515))OR (PEEK (516) * 128)
30 IF A = 255 OR (A<>66 AND A<>73 AND A<>194 AND A<>201)
    THEN 20
40 IF A = 66 THEN PRINT " ↓←*← "; : GOTO 20
50 IF A = 73 THEN PRINT " *← "; : GOTO 20
60 IF A = 194 THEN PRINT " ↑←*← "; : GOTO 20
70 IF A = 201 THEN PRINT " ←←*← "; : GOTO 20

```

The second will produce a "perfect resolution" sine wave down the paper on a Commodore 3022 tractor feed printer:

```

10 DIMA$ (5)
20 FORI = 0 TO 5
30 FORJ = 0 TO 5
40 READA:A$ (I) = A$ (I) + CHR$ (A)
50 NEXTJ,I
60 DATA64,0,0,0,0,0
65 DATA0,64,0,0,0,0
70 DATA0,0,64,0,0,0
75 DATA0,0,0,64,0,0
80 DATA0,0,0,0,64,0
85 DATA0,0,0,0,0,64
100 FORI=0 TO 2 * π STEP π / 100
105 B=(SIN(I)+1)*230
110 OPEN6,4,6:PRINT# 6,CHR$ (5)
120 OPEN5,4,5:PRINT# 5,A$ ((B/6-INT(B/6))*6)
130 OPEN4,4,0:PRINT# 4,TAB(INT(B/6));CHR$ (254)
140 CLOSE4:CLOSE5:CLOSE6:NEXTI

```

* * * * *

The letter below, originally came to us as a printout, presumably from Mr. Herbert's test processor. Regrettably, the text would not re-print and had to be re-typed. The routine is not as daft as it may seem!

Dept. of Animal Services
Ninewells Hospital
Dundee DD2

31st August 1979

Dear Sir,

You may like to have the enclosed short program for the 'Newsletter'. It eliminates itself!

This is not just a joke but can be used e.g. to eliminate instructions before using a text processor so as to free working space for strings. In that case line 7 would be a 'GETA\$' loop. Two or more, programs i.e. lines 8 & 9 can be placed in sequence so as to eliminate more lines.

My printer doesn't do graphics so the control characters have been represented as follows:

Clear screen = cls
Cursor down = d
Cursor up = ^
These are underlined in ink in the print-out submitted.

The program works as follows:

Line 8 prints the figures 1 to 9 down the screen and puts 9 'returns' into the keyboard buffer (poked into locations 527, 528 etc.)

Line 9 puts an additional 'return' into the buffer (536 by now) so that the instruction 'RUN100' is covered. This gets the program going again after 'END'. Three more 'Cursor ups' are printed than the number of lines to be deleted. Nine lines in this case, so 12 'Cursor ups'. Then the number of items in the buffer are put into location 525, and they are fired off by 'END'.

Yours sincerely,

W. J. Herbert

```
1  REM
2  REM
3  REM
4  REM
5  REM
6  REM
7  REM
8  PRINT"clsdd":FORI=1TO9:PRINT:POKE526+I,13:NEXT
9  POKE536,13:PRINT"RUN100^^^^^":POKE525,10:END
100 PRINT"clsELIMINATED!"
```

* * * * *

To simplify the use of our 2040 disk, we are making available a program called: "WEDGE - DOS SUPPORT PROGRAM". This adds several commands to the Pet in direct mode and simplifies several disk procedures. Dealers already have copies and it is being included with future deliveries of floppy disk units.

DOS SUPPORT PROGRAM

The purpose of this program is to aid the CBM or PET 2001 User in operating the 2040 Dual Floppy Disk System. This instruction sheet has been written with the assumption that the reader has a working knowledge of the 2001 series and the 2040.

NOTE: This program has been placed in the public domain. Please refer all comments and suggestions to the Editor.

The normal method with which the PET communicates with an IEEE Buss device is by the BASIC commands OPEN, PRINT, GET, INPUT and CLOSE. These statements are somewhat verbose in nature and therefore more prone to operator error. There is also the limitation that INPUT and GET cannot be used in direct mode due to shared buffer areas. These conditions are easily handled with the DOS SUPPORT PROGRAM.

DOS SUPPORT PROGRAM may be loaded (saved) as if it were a normal BASIC program. Note should be made of the fact that the 2040 has a special load file name '*' which if used immediatly after power up (reset) executes the following:

1. Initializes Drive 0
2. Loads the first file on that drive

Thus if the command LOAD "*",8 is executed and the DOS SUPPORT Program is the first directory entry it will be re-loaded. When the DOS SUPPORT Program is executed it re-locates itself up into the highest available RAM memory locations, links into the CHRGET routine and adjusts BASIC's top of memory pointer down. This technique uses about 350 bytes of the Users memory but normal machine operations may proceed without having to reload the DOS SUPPORT Program until such time that a system reset is performed.

The DOS SUPPORT Program functions by capturing the data that the PET operating system passes to BASIC, before the interpreter has a chance to spare it. Thus we can look for Key (escape) characters and process the disk command which follows without the use or knowledge of the BASIC interpreter.

There are four key characters that are recognized by the DOS SUPPORT Program. They will be processed only when they are found in column one of an input line, otherwise a SYNTAX ERROR will occur.

DOS SUPPORT KEY CHARACTERS

- @ or > - Passes commands to the Disk.
- / - LOAD's a program.
- ↑ - LOAD's and RUN's a program.

The greater than symbol when used preceding a 2040 Disk command, passes that command directly to the Floppy Disk System. See the following examples.

Thus:
>IØ
is the same as:
PRINT#15,"IØ"
and:
>SØ:FILE1
is equal to:
PRINT#15,"SØ:FILE1"

As you can see the > symbol is a substitute for the PRINT#15 statement. Remember that an OPEN statement is required before a PRINT may be executed but no OPEN is required for the DOS SUPPORT Program.

The second function of the > command is the directory list command. As you know the directory of a minidisk can be loaded with a LOAD"\$Ø",8. This LOAD will destroy any program you might have in memory. To avoid the destruction of the current program the DOS SUPPORT program prints the directory on the screen.

To avoid possible directory scrolling, you may depress the SPACE key to stop the listing of a directory. Depress any key to continue the listing - or you may depress the RUN/STOP key to stop the directory listing and return to BASIC.

>\$Ø

Means - Display the entire directory of Drive Ø

>\$1:Q*

Means - Display the directory entries of all files on Drive 1 that have names starting with the letter Q.

The third function of the > command is the error channel interrogation feature. The error channel is read by typing a > followed immediately by a RETURN. This is equivalent to the following program segment.

```
10 OPEN 15,8,15
20 INPUT#15,ER,MSG$,DRV,SEC
30?ER",MSG$","DRV","SEC
```

For Users that have the CBM Model Business Keyboard the "@" key may be used in place of the > for key entry convenience. This eliminates shifting for this command.

The LOAD / and LOAD-RUN + command characters operate the same as their BASIC counterparts only with a simplified syntax as follows: /WUMPUS

- This command will load the program file WUMPUS. Both drives will be searched if required.

†1:COPY DISK FILES

- This command will load the program COPY DISK FILES from Drive 1 (if it is there) and execute it.

The following requirements and limitations are placed on the DOS SUPPORT Program User.

1. The DOS SUPPORT commands may only be used in the direct mode.
2. Programs using GET or INPUT should disable the DOS SUPPORT program with the following command:

POKE 1022,128

Then, prior to program exit, DOS SUPPORT can be restored by using another POKE statement:

POKE 1022,08

3. The DOS SUPPORT Program can be disabled/restored on the direct mode as shown above, or may be disabled easily by typing:

>K and depressing RETURN

```
5 SYS2222
10 PRINT"J"TAB(11)""_
20 PRINTTAB(11)"" PET DOS SUPPORT "
30 PRINTTAB(14)""NOW LOADED"
40 PRINTTAB(9)"" COMMANDS FOLLOWING"
50 PRINTTAB(7)""A > OR @ IN COLUMN 1 WILL"
60 PRINTTAB(9)""BE PASSED TO THE DISK."
70 PRINTTAB(7)""CMD      DESCRIPTION"
140 PRINTTAB(7)""$      DIRECTORY BOTH DRIVES
150 PRINTTAB(7)""$0     DIRECTORY DRIVE 0
160 PRINTTAB(7)""$1     DIRECTORY DRIVE 1"
180 PRINTTAB(7)"" ALL 2040 COMMANDS MAY BE
190 PRINTTAB(7)""ENTERED AS IF THEY WERE IN
200 PRINTTAB(7)""A PRINT# STATEMENT.
220 PRINTTAB(11)""@SPECIAL COMMANDS
230 PRINTTAB(7)""L      LOAD A PROGRAM
240 PRINTTAB(7)""R      RUN A PROGRAM
250 PRINT"  SPECIAL COMMANDS START IN COL 1 AND
260 PRINT"ARE FOLLOWED BY A 2040 FILENAME.
270 NEW
```

PETDOS4.0.

LINE#	LOC	CODE	LINE
0001	0000		;*****
0002	0000		;*
0003	0000		;* PET DOS SUPPORT
0004	0000		;*
0005	0000		;* 04-27-79
0006	0000		;*
0007	0000		;* BOB FAIRBAIRN
0008	0000		;*
0009	0000		;*****
0010	0000		;*
0011	0000		;* VERSION 3.1 6/14/79
0012	0000		;* ADD @ PROMPT FOR BUSINESS
0013	0000		;* KEYBOARD. ADD STOP KEY CHECK
0014	0000		;* IN DIRECTORY PRINT. ADD
0015	0000		;* HALT IN DIRECTORY PRINT
0016	0000		;*
0017	0000		;* VERSION 3.2 7/2/79
0018	0000		;* FOR (-04) ROM
0019	0000		;* WITH LOAD ADDRESS ONE OFF
0020	0000		;* BYTE LOW.
0021	0000		;*
0022	0000		;* VERSION 3.3 7/2/79
0023	0000		;* ADD STACK LOOKUP FOR
0024	0000		;* ACTIVATION.
0025	0000		;*
0026	0000		;* VERSION 4.0 7/5/79
0027	0000		;* ADD CONTROL FOR CMD DURING
0028	0000		;* A DIRECTORY LISTING.
0029	0000		;*
0031	0000		;
0032	0000		;BASIC VARIABLES USED
0033	0000		;
0034	0000	VERCK	=\$9D ;VERIFY FLAG
0035	0000	SAL	=\$C7 ;INDIRECT POINTER LO
0036	0000	SAH	=\$C8 ;HI
0037	0000	WSW	=\$B3 ;UNUSED FLAG (BASIC)
0038	0000	CNTDN	=\$BA ;SAVE AREA
0039	0000	GRBTOP	=\$5C ;INDIRECT POINTER
0040	0000	MEMSIZ	=\$34 ;POINTER TO TOP MEM
0041	0000	TXTPTR	=\$77 ;POINTER TO BUF
0042	0000	SPERR	=\$10 ;EOI ERROR BIT
0043	0000	BUF	=\$0200 ;BASIC INPUT BUFFER
0044	0000	SATUS	=\$96 ;STATUS BYTE
0045	0000	SA	=\$D3 ;SECONDARY ADDRESS
0046	0000	FA	=\$D4 ;PRIMARY ADDRESS
0047	0000	LA	=\$D2 ;LOGICAL DEVICE #
0048	0000	FNLEN	=\$D1 ;FILE NAME LENGTH
0049	0000	FNAADR	=\$DA ;FILE NAME ADDRESS
0050	0000	EAR	=\$C9 ;END ADDR LO
0051	0000	EAH	=\$CA ;HI

0052	0000	DFL10	=\$B0	; DEFAULT OUTPUT DEV.	
0053	0000	VARTAB	=\$2A	; END OF BASIC PGM.	
0054	0000	TMP2	=\$FD	; TEMP VARIABLE	
0055	0000	;			
0056	0000	; PROGRAM VARIABLES			
0057	0000	;			
0058	0000	CR	=\$0D	; SYMBOLIC CARRIAGE RETURN	
0059	0000	FLAG	=WSW	; BYTE USED AS A FLAG	
0060	0000	PIAK	=\$E812	; KEYBOARD I/O PORT	
0061	0000	CMDLN	=CMDEND-CMDI	; LENGTH OF RELOCATE	
0063	0000	;			
0064	0000	; PET ROUTINES USED			
0065	0000	;			
0066	0000	LINPRT	=\$DCD9	; PRINT LINE #	
0067	0000	SPMSG	=\$F315	; SEND A MESSAGE	
0068	0000	LD15	=\$F322	; LOAD ROUTINE	
0069	0000	TWAIT	=\$F8E6	; WAIT FOR STOP KEY	
0070	0000	CHRGET	=\$70	; INPUTS CHARACTERS	
0071	0000	CHRGOT	=\$76	; GET LAST CHAR	
0072	0000	NEWSTT	=\$C6C4	; NEW STATEMENT EXEC	
0073	0000	PRT	=\$E3D8	; PRINT A CHARACTER	
0074	0000	LISTN	=\$F0B8	; SEND LISTEN	
0075	0000	SECND	=\$F128	; SEND SA	
0076	0000	CIOUT	=\$F16F	; SEND CHARACTER	
0077	0000	UNLSN	=\$F183	; UN LISTEN	
0078	0000	ACPTR	=\$F18C	; GET A CHARCATER	
0079	0000	TALK	=\$F0B6	; SEND TALK	
0080	0000	OPENI	=\$F466	; OPEN FILE	
0081	0000	FCLOSE	=\$F2AE	; CLOSE FILE	
0082	0000	READY	=\$C389	; REENTER BASIC	
0083	0000	RUNC	=\$C572	; CLEAR VARIABLES	
0084	0000	LNKPRG	=\$C442	; LINK BASIC LINES	
0085	0000	UHTLK	=\$F17F	; UN TALK	
0086	0000	STXTPT	=\$C5A7	; SET START TEXT POINTER	
0087	0000	CHKIN	=\$F770	; CHECK IN	
0088	0000	CHKOUT	=\$F78C	; CHECK OUT	
0089	0000	CLRCHN	=\$FFCC	; CLEAR CHANNEL	
0090	0000	BASIN	=\$FFCF	; BASIC IN	
0091	0000	STOP1	=\$F301	; CHECK FOR STOP KEY	
0092	0000	BSOUT	=\$FFI2	; BASIC OUT	
0093	0000	FOPEN	=\$F524	; FILE OPEN	
0094	0000	LI209	=\$F3E6	; LOAD ERROR	
0096	0000	;			
0097	0000	; WEIGE IN ROUTINE WITH THE			
0098	0000	; COMMAND PARSER AND EXECUTION			
0099	0000	;			
0100	0000	*=\$0700			
0101	0700	;			
0102	0700	EA	CMD	NOP	; THROWN AWAY
0103	0701	E6 77		INC TXTPTR	; BUMP POINTER
0104	0703	I0 02		BNE WG100	
0105	0705	E6 78		INC TXTPTR+1	
0106	0707	86 B3	WG100	STX WSW	; SAVE X IN WSW

0107	0709	BA	TSX	;GET STACK POINTER
0108	070A	BD 01 01	LDA \$0101,X	;WERE WE CALLED BY MAIN
0109	070D	C9 9B	CMP #\$9B	;NO...
0110	070F	D0 3A	BNE NOMAIN	;MAYBE?
0111	0711	BD 02 01	LDA \$0102,X	
0112	0714	C9 C3	CMP #\$C3	
0113	0716	D0 33	BNE NOMAIN	;NOT THERE...
0114	0718	A5 77	LDA TXTPTR	;FIRST COLUMN
0115	071A	D0 20	BNE WG997	;GET OUT NOT FIRST CHR
0116	071C	A5 78	LDA TXTPTR+1	
0117	071E	C9 02	CMP #>BUF	;IN BUFFER?
0118	0720	B0 26	BNE WG997	
0119	0722			
0120	0722	A0 00	WG110 LDY #0	;Y IS BUF INDEX
0121	0724	84 B3	STY FLAG	;FLAG SET FOR DIR
0122	0726	B1 77	LDA (TXTPTR),Y	
0123	0728	C9 3E	CMP #>	;COMMAND PROMPT?
0124	072A	F0 11	BEQ WG115	;YES...
0125	072C	C9 40	CMP #'@	;BUSINESS KEYBOARD PROMPT
0126	072E	F0 0D	BEQ WG115	;YES...
0127	0730	C8	INY	
0128	0731	85 B3	STA FLAG	;SET FLAG FOR LOAD
0129	0733	C9 2F	CMP #'/'	;LOAD PROMPT
0130	0735	F0 63	BEQ DODIR	
0131	0737	C9 5E	CMP #94	;CHECK FOR ARROW
0132	0739	F0 5F	BEQ DODIR	
0133	073B	D0 0B	BNE WG997	
0134	073D	C8	WG115 INY	
0135	073E	B1 77	LDA (TXTPTR),Y	
0136	0740	F0 32	BEQ RDERR	;READ ERROR CHANNEL
0137	0742	C9 24	CMP #'\$;DIRECTORY?
0138	0744	F0 54	BEQ DODIR	;YES
0139	0746	D0 08	BNE NOTDIR	
0140	0748	4C 76 00	WG997 JMP CHRGOT	
0141	074B	A6 B3	NOMAIN LDX WSW	;RESTORE .X AND
0142	074D	4C 76 00	JMP CHRGOT	;RETURN TO CHRGOT
0144	0750			
0145	0750			; SEND COMMAND TO DISK
0146	0750			
0147	0750	A9 08	NOTDIR LDA #8	;GET DEVICE ADDRESS
0148	0752	85 D4	STA FA	
0149	0754	A9 6F	LDA #\$6F	;SECONDARY ADDRESS 15
0150	0756	85 D3	STA SA	
0151	0758	20 BA F0	JSR LISTN	
0152	075B	A5 D3	LDA SA	
0153	075D	20 28 F1	JSR SECND	;SEND SECONDARY ADDR
0154	0760	E6 77	BUMP INC TXTPTR	
0155	0762	A0 00	LDY #0	;INDEX=0
0156	0764	B1 77	LDA (TXTPTR),Y	;GET THE FIRST CHARACTER
0157	0766	F0 06	BEQ WG120	;ZERO IS LAST CHAR
0158	0768	20 6F F1	JSR CIOUT	;SEND THE CHAR
0159	076B	B8	CLV	
0160	076C	50 F2	BVC BUMP	;MORE
0161	076E			
0162	076E	20 63 F1	WG120 JSR UNLSN	;UN LISTEN

0163	0771	B8	CLV	
0164	0772	50 23	BVC WG998	
0165	0774		;	
0166	0774		READ THE ERROR CHANNEL	
0167	0774		;	
0168	0774	84 77	RIDERR STY TXTPTR	:FIX POINTER
0169	0776	A9 08	LDA #8	:SET FA
0170	0778	85 D4	STA FA	
0171	077A	20 B6 F0	JSR TALK	
0172	077D	A9 6F	LDA #\$6F	:COMMAND CHANNEL SA
0173	077F	85 D3	STA SA	
0174	0781	20 28 F1	JSR SECND	:SEND SA
0175	0784	20 8C F1	WG140 JSR ACPTR	:GET BYTE FROM DISK
0176	0787	C9 0D	CMP #CR	
0177	0789	F0 06	BEQ WG130	
0178	078B	20 D8 E3	JSR PRT	:PRINT BYTE TO SCREEN
0179	078E	B8	CLV	
0180	078F	50 F3	BVC WG140	:LOOP FOR MORE
0181	0791	20 D8 E3	WG130 JSR PRT	:PRINT CR
0182	0794	20 7F F1	JSR UNTLK	:UN TALK
0183	0797	4C 76 00	WG998 JMP CHRGOT	:DONE WITH CMD
0185	079A		;	
0186	079A		PRINT THE DIRECTORY	
0187	079A		;	
0188	079A	C8	DODIR INY	:GET LENGTH OF CMD
0189	079B	B1 77	LDA (TXTPTR),Y	
0190	079D	D0 FB	BNE DODIR	
0191	079F	88	DEY	
0192	07A0	84 D1	STY FNLEN	:SET LENGTH (-1)
0193	07A2	A9 01	LDA #<BUF+1	:FILE NAME ADDRESS
0194	07A4	85 DA	STA FNADR	
0195	07A6	A9 02	LDA #>BUF	
0196	07A8	85 DB	STA FNADR+1	
0197	07AA	A9 08	LDA #8	:DEVICE ADDRESS
0198	07AC	85 D4	STA FA	
0199	07AE	A5 B3	LDA FLAG	: 0 MEANS DIR
0200	07B0	D0 53	BNE LORDB	:DO A LOAD
0201	07B2	A5 D2	LDA LA	:SAVE LA
0202	07B4	85 B3	STA WSW	
0203	07B6	A5 B0	LDA DFLTO	:SAVE DFLTO
0204	07B8	85 BA	STA CNTIN	
0205	07BA	A9 60	LDA #\$60	:SECONDARY ADDR
0206	07BC	85 D3	STA SA	
0207	07BE	A9 0E	LDA #14	:OPEN THE FILE
0208	07C0	85 D2	STA LA	
0209	07C2	20 83 F1	JSR UNLSH	:DON'T LISTEN TO FLOPPY
0210	07C5	20 24 F5	JSR FOPEN	
0211	07C8	A9 00	LDA #0	
0212	07CA	85 96	STA STATUS	:SET STATUS TO 0
0213	07CC	A0 03	LDY #\$03	:LOOP THREE TIMES
0215	07CE	84 D1	WG220 STY FNLEN	:SAVE NEW COUNT
0216	07D0	A2 0E	LDX #14	:DISK CHANNEL
0217	07D2	20 70 F7	JSR CHKIN	
0218	07D5	20 CF FF	JSR BASIN	
0219	07D8	85 F0	STA TMP2	
0220	07DA	A4 96	LDY STATUS	:CHECK STATUS
0221	07DC	D0 29	BNE WG235B	:BAD STATUS
0222	07DE	20 CF FF	JSR BASIN	
0223	07E1	85 FE	STA TMP2+1	
0224	07E3	A4 96	LDY STATUS	:CHECK STATUS
0225	07E5	D0 20	BNE WG235B	
0226	07E7	A4 D1	LDY FNLEN	:MORE TO DO?
0227	07E9	88	DEY	
0228	07EA	D0 E2	BNE WG220	:NOT DONE YET
0229	07EC	20 CC FF	JSR CLRCHN	:CLEAR CHANNEL

0230	07EF	A6 B8	LDX CNTDN	; CHECK DFLTO FOR SCREEN
0231	07F1	E8 03	CPX #3	
0232	07F3	F0 05	BEQ *+7	
0233	07F5	A6 B3	LDX WSW	; OPEN THE PRINT CHANNEL
0234	07F7	20 BC F7	JSR CHKOUT	
0235	07FA	A6 FD	LDX TMP2	
0236	07FC	A5 FE	LDA TMP2+1	
0237	07FE	20 D9 DC	JSR LINPRT	; PRINT LINE NUMBER
0238	0801	A9 20	LDA #'	; PRINT A SPACE
0239	0803	D0 06	BNE SKIPB	; SKIP OVER BRANCHES
0240	0805	D0 6C	LOADB BNE LOAD	; (JMP)
0241	0807	D0 5D	WG235B BNE WG230	; (JMP)
0242	0809	D0 C3	WG220B BNE WG220	; (JMP)
0243	080B	20 D2 FF	SKIPB JSR BSOUT	
0244	080E	20 CC FF	JSR CLRCHN	
0245	0811	A2 0E	WG250 LDX #14	; DISK CHANNEL
0246	0813	20 78 F7	JSR CHKIN	
0247	0816	20 CF FF	JSR BASIN	
0248	0819	48	PHA	
0249	081A	20 CC FF	JSR CLRCHN	
0250	081D	68	PLA	
0251	081E	A6 96	LDX STATUS	
0252	0820	D0 44	BNE WG230	; BAD
0253	0822	C9 00	CMP #'0	; EOL
0254	0824	F0 26	BEQ WG240	
0255	0826	A6 B8	LDX CNTDN	; CHECK DFLTO FOR SCREEN
0256	0828	E0 03	CPX #3	
0257	082A	F0 05	BEQ *+7	
0258	082C	A6 B3	LDX WSW	
0259	082E	20 BC F7	JSR CHKOUT	
0260	0831	20 D2 FF	JSR BSOUT	
0261	0834	20 CC FF	JSR CLRCHN	
0262	0837			
0263	0837			; CHECK FOR STOP KEY AND PAUSE
0264	0837			
0265	0837	20 01 F3	JSR STOP1	; STOP KEY
0266	083A	F0 2A	BEQ WG230	; YES...
0267	083C	20 E4 FF	JSR \$FFE4	; GET A CHAR FROM KEYBOARD
0268	083F	F0 D0	BEQ WG250	; NOTHING...
0269	0841	C9 20	CMP #\$20	; SPACE BAR?
0270	0843	D0 CC	BNE WG250	; NO...
0271	0845	20 E4 FF	WG255 JSR \$FFE4	; ANY KEY STARTS
0272	0848	F0 FB	BEQ WG255	
0273	084A	D0 C5	BNE WG250	; (JMP)
0274	084C			
0275	084C	A9 0D	WG240 LDA #CR	
0276	084E	A6 B8	LDX CNTDN	; CHECK DFLTO FOR SCREEN
0277	0850	E0 03	CPX #3	
0278	0852	F0 05	BEQ *+7	
0279	0854	A6 B3	LDX WSW	
0280	0856	20 BC F7	JSR CHKOUT	
0281	0859	20 D2 FF	JSR BSOUT	
0282	085C	20 CC FF	JSR CLRCHN	
0283	085F	20 83 F1	JSR UNLSN	
0284	0862	A0 02	LDI #\$02	; DO TWICE
0285	0864	D0 A3	BNE WG220B	
0286	0866			
0287	0866			; CLOSE FLOPPY AND RETURN
0288	0866			
0289	0866	20 CC FF	WG230 JSR CLRCHN	
0290	0869	A9 0E	LDA #14	; CLOSE FLOPPY
0291	086B	20 AE F2	JSR FCLOSE	
0292	086E	68	PLA	; CLEAN UP THE STACK
0293	086F	68	PLA	
0294	0870	4C 89 C3	JMP READY	; RETURN "READY"

0296	0873		; LOAD A FILE			
0297	0873		;			
0298	0873		;			
0299	0873	A9 00	LOAD	LDA #0		
0300	0875	85 96		STA STATUS	;CLEAR STATUS	
0301	0877	85 9D		STA VERCK	;LOAD NOT VERIFY	
0302	0879	20 22 F3		JSR LD15	;LOAD A PROGRAM	
0303	087C	A5 96		LDA STATUS		
0304	087E	29 10		AND #SPERR	;CHECK STATUS (EOI OK)	
0305	0880	D0 28		BNE LDERR		
0306	0882	AD 84 F3		LDA \$F384	;CHECK FOR (-04) ROM	
0307	0885	30 06		BMI LOAD1	;NOT (-04)....	
0308	0887	E6 C9		INC EAL	;FIX THE LOAD (-04) ROM	
0309	0889	D0 02		BNE LOAD1		
0310	088B	E6 CA		INC EAH		
0311	088D	A5 CA	LOAD1	LDA EAH	;SET BASIC'S POINTERS	
0312	088F	85 28		STA VARTAB+1		
0313	0891	A5 C9		LDA EAL		
0314	0893	85 2A		STA VARTAB		
0315	0895	20 72 C5		JSR RUNC	;FIX POINTERS	
0316	0898	20 42 C4		JSR LNKPRG	;FIX LINKS	
0317	089B	A5 B3		LDA FLAG	;CHECK FOR LOAD OR RUN	
0318	089D	C9 2F		CMP #1/	;LOAD ?	
0319	089F	D0 03		BNE WG300	;NO...	
0320	08A1	4C 89 C3		JMP READY	;LOAD RETURN TO BASIC	
0321	08A4	20 A7 C5	WG300	JSR STXTPT	;SET TXTPT FOR RUN	
0322	08A7	4C C4 C6		JMP NEWSTT	;RUN PROGRAM	
0323	08AA	4C E6 F3		LDERR	JMP LD209	;PRINT "LOAD ERROR"
0324	08AD			CMDEND		

0326	08AD		; THIS ROUTINE POKE'S TOP OF MEMORY		
0327	08AD		; DOWN RELOCATES THE PARSER AND		
0328	08AD		; SETS THE WEDGE		
0329	08AD		;		
0330	08AD		POKE	LDA MEMSIZ	;POKE TOP DOWN
0331	08AD	A5 34		CLC	;MINUS ONE
0332	08AF	18		SBC #<CMDLN	
0333	08B0	E9 A9		STA MEMSIZ	
0334	08B2	85 34		LDA MEMSIZ+1	
0335	08B4	A5 35		SBC #>CMDLN	
0336	08B6	E9 01		STA MEMSIZ+1	
0337	08B8	85 35			
0338	08BA		; MOVE THE CODE		
0339	08BA		MOVE	LDY #\$01	;SET UP FROM ADDR
0340	08BA			LDA #<CMD	
0341	08BA	A9 01		STA SAL	
0342	08BC	A9 00		LDA #>CMD	
0343	08BE	85 07		STA SAH	
0344	08C0	A9 07		LDA MEMSIZ	;SET UP TO RIDE
0345	08C2	85 08		STA GRBTOP	
0346	08C4	A5 34		LDA MEMSIZ+1	
0347	08C6	85 50		STA GRBTOP+1	
0348	08C8	A5 35		STA GRBTOP+1	
0349	08CA	85 50		LDA (SAL),Y	;RELOCATE
0350	08CC	B1 07	MOV1	STA (GRBTOP),Y	
0351	08CE	91 50		INY	
0352	08D0	C8		BNE MOV1	
0353	08D1	D0 F9		INC GRBTOP+1	
0354	08D3	E6 5D		INC SAH	
0355	08D5	E6 C8			

0356	08D7	R5 C8		LDA SAH		
0357	08D9	C9 08		CMP #CMDEND		
0358	08DB	F0 02		BEG MOV2		
0359	08DD	B0 04		BGS WEDGE		
0360	08DF	A0 00	MOV2	LDY #0		
0361	08E1	F0 E9		BEG MOV1		
0362	08E3			;		
0363	08E3			:WEDGE INTO BASIC		
0364	08E3			;		
0365	08E3	R9 4C	WEDGE	LDA #\$4C	JUMP INSTRUCTION	
0366	08E5	85 70		STB CHRGET		
0367	08E7	R4 34		LDY MEMSIZ		
0368	08E9	A6 35		LDX MEMSIZ+1		
0369	08EB	C8		INY		
0370	08EC	I0 01		BNE WEDGE1		
0371	08EE	E8		INX		
0372	08EF	84 71	WEDGE1	STY CHRGET+1		
0373	08F1	86 72		STX CHRGET+2		
0374	08F3	60		RTS		
0375	08F4			.END		

ERRORS = 0000

SYMBOL TABLE

SYMBOL	VALUE						
ACPTR	F180	BASIN	FFCF	BSOUT	FFD2	BUF	0200
BUMP	0760	CHKIN	F770	CHKOUT	F780	CHRGET	0070
CHRGET	0076	CIOUT	F16F	CLRCHN	FFCC	CMD	0700
CMDEND	08AD	CMDLN	01AD	CNTIN	00BR	CR	0000
DFLTO	00B0	DODIR	079A	EAH	00CA	ERL	00C9
FA	00D4	FCLOSE	F2AE	FLAG	00B3	FNAIR	00DA
FNLEN	00D1	FOPEN	F524	GRBTOP	005C	LA	00D2
LD15	F322	LI209	F3E6	LDERR	08AA	LINPRT	DCD9
LISTN	F0BA	LNKPRG	C442	LOAD	0873	LOAD1	0880
LOADB	0805	MEMSIZ	0034	MOV1	080C	MOV2	081F
MOVE	08BA	NEWSTT	C6C4	NOMAIN	074B	NOTDIR	0750
OPENI	F466	PIAK	E812	POKE	08AD	PRT	E3D8
RDERR	0774	READY	C3E9	RUNC	C572	SA	00D3
SAH	00C8	SAL	00C7	SATUS	0096	SECND	F128
SKIPB	080B	SPERR	0010	SPMSG	F315	STOP1	F301
STXTPT	C5A7	TALK	F0B6	TMP2	00FD	TWAIT	F8E6
TXTPTR	0077	UNLSN	F183	UNTLK	F17F	VARTAB	002A
VERCK	009D	WEIDGE	08E3	WEIDGE1	08EF	WG100	0707
WG110	0722	WG115	073D	WG120	076E	WG130	0791
WG140	0784	WG220	07CE	WG220B	0809	WG230	0866
WG235B	0807	WG240	084C	WG250	0811	WG255	0845
WG300	08A4	WG997	0748	WG998	0797	WSW	00B3

END OF ASSEMBLY

6502 MICROPROCESSOR ASSEMBLER FOR CBM PET 2001

ASMPAC8 is a powerful combined machine code and Basic Assembler program for MK 1 8K PETs. ASMPAC8 includes a TEXT EDITOR an ASSEMBLER and a DISASSEMBLER with a host of useful features to run in 7167 BYTES FREE. The Assembler will (1) Test Assemble to allow errors to be found (2) Assemble directly into free memory or (3) Assemble object code files for later processing by AUTOLOAD8 or AUTOSTRINGS8. ASMPAC8 is an ideal tool for beginners as the Test Assembly facility allows errors to be detected and corrected immediately with the Editor. ASMPAC8 allows the more experienced programmer to produce programs rapidly as the use of Files for Assembly Text is optional for short programs.

ASMPAC8 is a practical development system and has been designed for ease of use.

ASMPAC8 complete with the ASMPAC manual 25 pounds.

AUTOLOAD8 is a useful program to convert ASMPAC8 object code files into free standing machine code programs that may be SAVED, VERIFIED, LOADED and RUN just like a Basic program but without the need for a basic support program.

AUTOLOAD available only with ASMPAC8 6 pounds.

AUTOSTRINGS8 will convert ASMPAC8 object code files into Basic strings which may then be transferred into the second cassette buffer for execution. An ideal way to add short routines to be called from Basic by the 'SYS' or 'USR' functions. The program includes the string conversion routine and an automatic removal of the file reading section.

AUTOSTRINGS8 available only with ASMPAC8 5 pounds.

ASMPAC16/32 for the new PETs with MK2 ROMS now available with expanded Editor, Assembler and debugging routines.

ASMPAC16/32 28.50

ORDER STATING IF 8K 16K or 32K PET AND INCLUDING 50p POST AND PACKING FROM
J C LEMAN, 47 LONDON ROAD, SOUTHBOROUGH, TUNBRIDGE WELLS, KENT

ASMPAC8 and it's support programs are RECOMMENDED BY CBM.

Book Review

"6502 APPLICATIONS BOOK" by Rodney Zaks

This book starts where "Programming the 6502" (reviewed in Issue No. 6) left off. Rodney Zaks is a well respected author of computing publications and has had considerable experience in training newcomers to the field.

The "6502 Applications Book" deals with all the Input/Output associated with 6502 based systems. As such, the book is particularly relevant to Pet users who wish to interface peripherals and 'home grown' devices to their computer.

A large amount of space at the beginning of the book is devoted to explaining the intricacies of the various support chips manufactured for the 6500 range of micro-processor. These include PIO's such as the 6520 for handling parallel data complete with hand shaking, and the more advanced Versatile Interface Adaptors like the 6522 which contains two programmable timers and a Shift Register for converting between serial and parallel data. The principles and operation of these devices are explained in detail from absolute basics and the text is interspersed with numerous diagrams, and worked examples. In addition there is a large number of test questions enabling you to keep a check on your understanding of each new concept.

The rest of the book deals with the practical side of interfacing and is illustrated with several diverse applications including; a traffic control system, a burglar alarm, a music generator and many others. All the assembly language routines (an understanding of A.L. programming is a pre-requisite) are written in a machine-independent form, for ease of application.

If you intend to interface any device to your Pet, or simply want to learn how a system like the Pet works internally, then this is probably one of the best books available on the subject.

The "6502 Applications Book", as with the previous book, is published by SYBEX and should be obtainable from your nearest computer bookshop.

* * * * *

"BASIC COMPUTER GAMES"

Edited by David H. Ahl

As the title suggests, this is a collection of computer games written in BASIC. The games come from 'Creative Computing' and have been collected and adapted largely by David Ahl (founder and publisher of C.C.). As all the programs are written in Microsoft 8K Basic they are completely compatible with the Pet and the listings can be entered straight into the keyboard and recorded for later use.

The book contains 101 games in total, ranging from old favourites including Lunar Landing and various card games, to some refreshingly original games such as Animal, and Poetry. On the whole, the games tend to be of the static type, requiring mental judgement rather than just reactions. The listings and documentation are presented in a tidy manner and on the whole, are unambiguous. A large number of anthropomorphic computer cartoons gives the publication an informal and 'friendly' image.

On the negative side, most of the listings could have been made considerably more efficient without loss of clarity. I found, however, that the programs were relatively easy to compress when entering them into the Pet - saving time and memory space. Also, Pet's graphic capabilities enable several of the games to be made more visually interesting with a little extra work.

Having said that, the book represents very good value for money at a published price of £4.50. It is distributed in the U.K. by:-

Transatlantic Book Service Limited
24 Red Lion Street
London WC1R 4PX.

* * * * *

Situation Vacant

Wanted PET Enthusiast

Wanted, an enthusiastic and knowledgeable exponent of the Commodore PET to help run Commodore's PET User Centre at Euston Road.

The above position is open to both male and female applicants.

Please apply in writing and enclose a Curriculum Vitae to:-

Ms. M. Newman,
PET User Centre,
360 Euston Road,
London.
NW1 3BL

Printout

1 Sorting Techniques Continued.

Several people have written to me following my re-publication of the SHELL-METZNER sort. All have pointed out that the QUICKSORT, by Hoare, is even quicker than the SHELL.

Working on the principle that it is easier to sort two small arrays rather than one large one, it chooses an element, and splits the array in two, placing smaller elements earlier and larger elements later in the array than the chosen element. It then operates on each of the smaller arrays in exactly the same way. Rather like a clerk sorting manually into smaller and smaller piles, the routine eventually produces sub-arrays which are in order and contain only one or two elements each. One final exchange in the case of the two element sub-array produces the sorted data.

QUICKSORT likes jumbled data. On average the sort time for N elements will be proportional to $N \cdot \log(N)$. The time for the SHELL will be N to the power $5/4$ and for the Bubble, N squared - on average.

Hoare suggests choosing the first element as the comparison item. However, when the array is ordered or nearly so, most other elements get placed on one side of it, creating a very small and a very large sub-array. In this case sort times approach that of the Bubble sort. One answer suggested by Harrington is to choose the middle array element for comparison. This works fine for ordered arrays but not for two ordered arrays joined end to end. He suggests that choosing an element at random is the way out of this problem.

You will see that where you are looking for the quickest way to sort data, it is necessary to consider the nature of that data and depending on how it is arranged, chose a suitable routine. If you can't predict the nature of the data, then probably the best compromise is the SHELL.

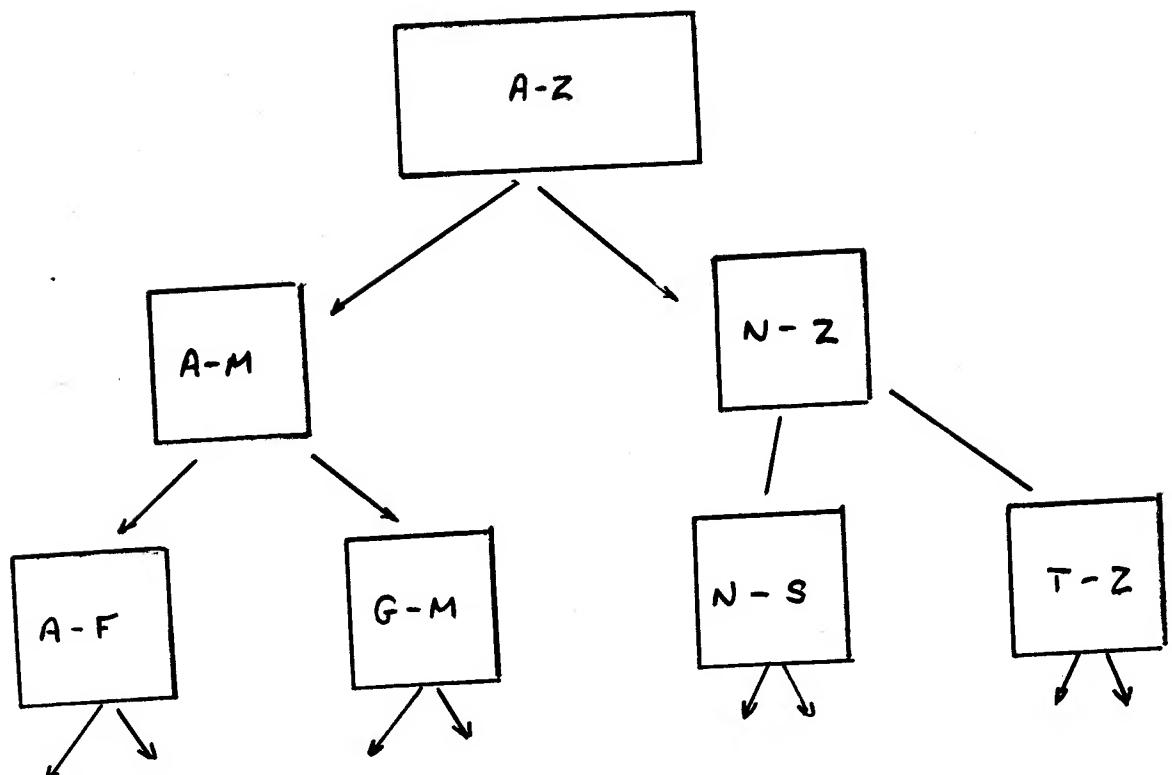
The following listing is taken from Harrington's article in "Micro Computing."

```
95 REM SET ARRAY SIZE
100 LET M=N
105 REM SET SPACING BETWEEN LIST MEMBERS
110 LET M=INT(M/2)
115 REM SEE IF DONE
120 IF M=0 THEN 300
130 LET J=1
140 LET K=N-M
145 REM I & L INDICATED ELMNTS TO BE MERGED
150 LET I=J
160 LET L=I+M
```

```

165 REM PERFORM MERGE
170 IFA(I)<A(L)GOTO240
180 LETT=A(I)
190 LETA(I)=A(L)
200 LETA(L)=T
210 LETI=I-M
220 IFI<1THEN240
230 GOTO160
235 REM MERGE NEXT TWO ELEMENTS
240 LETJ=J+1
250 IFJ<=KTHEN150
255 REM BEGIN MERGE OF RESULTANT LISTS
260 GOTO110
300REM END OF SORT

```



Etc.

DIAGRAMATIC REPRESENTATION OF 'QUICK SORT'

2. Symbolic BASIC 'assembler'.

2.1 Processing a 'listed' BASIC text.

In the last issue, I described a method of processing BASIC listed text on tape to produce a formatted listing. The technique can be extended to actually change the text contents, which can then be fed back into the memory as a proper but amended program using our old friend, the Templeton-Butterfield merge.

It is not a straightforward matter to change for example all occurrences of the variable T to become variable V, because of the problems in distinguishing variable T from variable CT or even PRINT.

2.2 A personal computer language.

A solution to the problem is to head each variable you type in with a special character such as ! which is not part of the normal BASIC syntax. The function of this header character is to say "here is the start of a variable.

This means that what you type in is not BASIC but will have to be converted using an 'assembly' program.

Taking matters further, one can type in explicit variable names headed by ! and ending with a space say, which BASIC cannot differentiate but your assembler program can. For example, one can use such variable names as !PAYNET , !PAYGROSS , and even !PAYDEDUCTIONS , which BASIC would either see as variable PA or complain because it finds the reserved word 'TI' or 'ON' hidden inside the name.

The way an assembler handles these psuedo variables is to create a table of true variable assignments, allocating real variable names A - ZZ as it meets each new psuedo- variable.

When the program has been assembled onto a tape or disk file, your assembler can then print out this table sorted into both psuedo and real variable order. Such a printout makes an excellent addition to the program documentation.

For example

```
Psuedo text
10 FOR !INDEX =!START TO!FINISH
20 PRINT!NAMES$ (!INDEX)
30 NEXT
```

becomes

```
10 FOR A=BTOC
20 PRINTA$(A)
30 NEXT
```

and the printed tables are

A=INDEX	FINISH=C
A\$=NAME\$	INDEX=A
B=START	NAME\$=A\$
C=FINISH	START=B

2.3 Jump to labels.

Taking the idea yet one more step, one can overcome the problem of having to type in GOTO, GOSUB and THEN without yet knowing what line number you are going to jump to.

The idea is to use label references and labels again headed by special characters, perhaps \ and '.

Again a table is formed of all labels and their line numbers. A second pass picks up label references and converts them to line numbers using the table.

For example

Your text

```
10 \START
20 INPUT"NUMBER 10 OR LESS";!NUMBER
30 IF!NUMBER >10THEN'ERROR
40 ?"YOUR NUMBER WAS 'OK'":END
50 \ERROR
60 ?"YOUR NUMBER WAS TOO HIGH"
70 GOTO'START
```

would assemble as

```
10 REM START
20 INPUT"NUMBER 10 OR LESS":A
30 IFA>10THEN50
40 ?"YOUR NUMBER WAS 'OK'":END
50 REM ERROR
60 ?"YOUR NUMBER WAS TO HIGH"
70 GOTO10
```

And once again your assembler would produce reference lists

ERROR 50	10 START
START 10	50 ERROR

Be warned that such assembly using tape and GET# is not fast and that your keyed in programs cannot be tested until they are assembled. However you will get considerable pleasure and satisfaction creating your own personal programming language and with large programs the technique may have some practical value.

3. BASIC competition number 4 "Wardrobe Removal"

This problem simulates one faced by those burly gentlemen who help you move house.

Write a BASIC program which asks for the dimensions of a box (right angled), then the height and width of a corridor and finally tells you whether the box will go round a right-angled turn in that corridor.

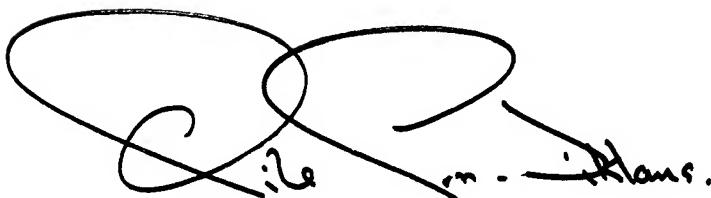
Entries will be judged on REM annotation as well as execution time. The winner will get f10 of Commodore software. Entries to me within 21 days of the official publication date of this issue please.

4. Results of BASIC competition number 3, "ARRAY ARRAY".

Yet again the formidable Mr J Clark of Watford emerges as the winner of competition no 3, closely followed by Mr K.D Armstrong of Edinburgh.

Mr Clark again provided copious notes. He developed the sort routines himself and came to the same type of solution as Hoare and Harrington.

```
1 I=0:Q=0:P=0:J=0:K=0:S=0:
2 N=255:DIMA(N),M(N),L(N),B(N)
2000 GOSUB3000:FORI=1TON:L(I)=A(P):
A(P)=B(P):P=M(P):NEXT
2010 GOSUB3000:FORI=1TON:B(I)=A(P):P=M(P):NEXT
2020 FORI=1TON:M(I)=B(I):NEXT
2030 J=1:K=1:FORI=1TON:P=L(J):Q=M(K):
IFP<QTHEN A(I)=P:J=J+1:NEXT:GOTO2050
2040 A(I)=Q:K=K+1:NEXT
2050 J=N:K=N:FORI=NTO1STEP-1:P=L(J):Q=M(K):
IFP QTHEN B(I)=P:J=J-1:NEXT:GOTO2070
2060 B(I)=Q:K=K-1:NEXT
2070 RETURN
3000 F=-15
3010 F=F+16:GOSUB4000:P=S:F=F+16:GOSUB4000:
Q=5:GOSUB5000
3020 IFF=170RF=810RF=1450RF=209THEN X=P:GOTO3010
3030 Q=X:GOSUB5000:IFF=490RF=177THEN Y=P:GOTO3010
3040 Q=Y:GOSUB5000:IFF=113THEN Z=P:GOTO3010
3050 Q=Z:GOSUB5000:RETURN
4000 S=F:M(F)=0:FORI=F+1TO F+15+(F=241)
4010 K=A(I):Q=I-1:IF K<A(Q) THEN 4060
4020 J=M(Q):IF J=0 THEN M(I)=0:GOTO4050
4030 IF K>A(J) THEN Q=J:GOTO4020
4040 M(I)=J
4050 M(Q)=I:NEXT:RETURN
4060 Q=S:IF K>A(Q) THEN 4020
4070 S=I:M(I)=Q:NEXT:RETURN
5000 I=P:IFA(Q)<A(I)THEN P=Q:Q=I:I=P
5010 J=M(I):IF J=0 THEN M(I)=Q:RETURN
5020 IFA(Q)<A(J)THEN M(I)=Q:I=Q:Q=J:GOTO5010
5030 I=J:GOTO5010
```



Petbits

A new training course entitled "MICROCOMPUTERS - Applications and Programming" is to be run at the Coventry Management Training Centre. Aimed primarily at businessmen, the course aims to demonstrate the capabilities of microcomputers and teach a working knowledge of BASIC, with hands-on experience on Commodore PETs.

The course lasts for three days (5th - 7th November 1979) and consists of lectures, demonstrations and teach-ins by qualified staff. The basic fee is £138.00 for the three days, including lunches and 15% VAT; hotel accommodation can be arranged at extra cost.

Although a large number of machines will be available on the course, we are advised that anyone able to bring his own PET will be eligible for a discount. For more information ring: 0926 36621.

* * * * *

NEW ENTRIES TO USERS DIRECTORY:

Gordon Bell
55 Belvedere Road, Hessle, Humberside. Tel: 0482 645724
Training, Consultancy and programming services.

Tina Björnstjerna Datatjänst
Brantingsgatan 50, S-115 35 Stockholm, Sweden.
Hardware, software for Pet.

J. E. Atlas
A De B Consultants, 3 Cromwell Place, London SW7.
Mechanical, Electrical, Civil, Structural Engineering.

* * * * *

The States-originated publication 'CURSOR', is now available in the U.K. through PETSOFT. Cursor comes on a C30 cassette, 10 times a year. At least five full size programs come on each cassette with one or more games programs that could be worth £8 elsewhere. None of the programs have previously been published in the U.K. In addition to the cassette the subscription includes CURSOR NOTES which cover the documentation for the programs and other Pet-related information.

One year's subscription to CURSOR costs £36.00 (£45.00 for overseas airmail). A sample issue may be obtained for £4.00 including p. & p. For more information telephone: 021 455 8585.

* * * * *